06. & 07. October 2010

OPEN SOURCE MONITORING CONFERENCE

OSMC ON NAGIOS

Java Monitoring und Troubleshooting

Rainer Jung, Geschäftsführer
kippdata informationstechnologie GmbH

© 2010 kippdata informationstechnologie GmbH

Agenda

- Aufwärmrunde
- Motivation
- Java Management Extensions (JMX)
- JDK-Tools Java 5 und 6
- Thread Dumps
- Diskussion!

Agenda

- Aufwärmrunde
- Motivation
- Java Management Extensions (JMX)
- JDK-Tools Java 5 und 6
- Thread Dumps
- Diskussion!

Aufwärmrunde – Wer bin ich?

- Wer bin ich?
  - Geschäftsführer kippdata informationstechnologie GmbH
    - Gründung 1998
    - 25 Mitarbeiter
  - Erfahrungshintergrund: Schwerpunkt Systemintegration
    - Hinzufügen von Produktionsqualitäten (Performance, Ausfallsicherung) zu (leider) meist schon fertigen Anwendungen
    - Troubleshooting (durch den ganzen Stack hindurch)

und auf dieser Basis dann ...

© 2010 kippdata informationstechnologie GmbH

Aufwärmrunde – Wer bin ich?

- Open Source-Leser
  - Problemanalyse
- Open Source-Contributor
  - Problembehebung
- Apache Tomcat-Committer und PMC-Mitglied
- Apache HTTP Server-Committer
- Apache APR-Committer
- mod_jk-Maintainer
- Member der Apache Software Foundation (ASF)

© 2010 kippdata informationstechnologie GmbH

Aufwärmrunde – Wer sind Sie?

- Wer sind Sie?
  - Wer gehört eher zu
    - Entwicklung?
    - Betrieb?
  - Wer trägt Verantwortung für Produktionsstabilität oder Fehleranalyse?
  - Wer hat mit 24x7x365-Anwendungen zu tun?
  - Wer musste schon Probleme debuggen, die erst in der Produktion beobachtet wurden?

© 2010 kippdata informationstechnologie GmbH

Aufwärmrunde – Wer sind Sie?

- Wer hat in der Produktion
  - Java 1.4.2
  - Java 5
  - Java 6
  - Eine nicht-Sun VM
- Wessen Code läuft nicht Standalone, sondern in einer Form von Container (Tomcat, Application Server, ...)?
- Wer macht schon Monitoring für Java-Anwendungen?
- Bei wem basiert HW-Kapazitätsplanung auf gemessenen Werten?
  - Gilt dies auch für das Sizing von SW-Komponenten (Pools etc.)?

© 2010 kippdata informationstechnologie GmbH

Aufwärmrunde – Wer sind Sie?

- Wer weiß, was ein MBean ist?
- Wer weiß, was ein Java Thread Dump ist?
  - Haben Sie schon ein Problem durch Thread Dumps gelöst?
- Wer hat schon einmal zu einem Open Source-Projekt beigetragen?
  - Patch, Doku, Problemanalyse
- Wer ist Projektmitglied bei einem Open Source-Projekt?

Aufwärmrunde – Wer sind Sie?

- Dieser Vortrag
  - Betrifft im wesentlichen Server-Anwendungen
  - Bezieht sich fast ausschließlich auf die Sun JVM
    - Java 5 und 6

© 2010 kippdata informationstechnologie GmbH

- Aufwärmrunde
- Motivation
- Java Management Extensions (JMX)
- JDK-Tools Java 5 und 6
- Thread Dumps
- Diskussion!

© 2010 kippdata informationstechnologie GmbH

Motivation – Monitoring

- Ziele von Monitoring
  - Fehlererkennung, Rot/Grün-Ampeln, Alarmierung
    - Automatische Erkennung und Meldung kritischer Fehler
    - Möglichst keine Fehlalarme
    - Möglichst Meldung der Wurzelursache
    - Meist aber nur gut verstandene Basiszustände und End-to-End
      - Filesystem voll, CPU ausgelastet
      - Anwendungs-Login, Test-Transaktion

Motivation – Monitoring

- Weitere Ziele von Monitoring
  - Kontinuierliche Sammlung von Laufzeitwerten
    - Pollen der Daten
    - Ablage der Daten
    - Verdichtung und Visualisierung
  - Wird benötigt
    - zur Analyse der Wurzelursache bei wiederkehrenden Problemen
    - für Kapazitätsmanagement (z.B. Anpassung Sizing)

© 2010 kippdata informationstechnologie GmbH

Motivation – Monitoring

- Die Betriebssicht
  - Den Betrieb interessiert meist nur die Meldung von Störungen (Incident Management)
  - Der Betrieb greift auf die Entwicklung zurück, wenn es um die Analyse komplexerer Probleme geht
- Die Entwicklungssicht
  - Zur erfolgreichen Analyse von Betriebsproblemen benötigen Sie Auswertungen über das Zeitverhalten der Systemkomponenten
  - Diese muss der Betrieb bereitstellen

© 2010 kippdata informationstechnologie GmbH

Motivation – Troubleshooting

- Troubleshooting:
  Die häufigsten technischen Probleme sind
  - Performance-Probleme
    - Schlechte Antwortzeiten
    - Schlechter Durchsatz
  - Stabilitäts-Probleme
    - Die Anwendung reagiert nicht mehr

Motivation – Troubleshooting

- Die häufigsten Problemursachen sind
  - Überlastete Backends
  - Locking-Probleme
  - Memory-Probleme (und schlechte GC-Einstellungen)
  - Falsches Sizing der eingesetzten Software-Komponenten
    - Pools, Caches, Timeouts
- Fast nie: Engpass CPU, Platte oder Netz

© 2010 kippdata informationstechnologie GmbH

Motivation – Troubleshooting

- Was wollen wir zur Laufzeit herausfinden?
  - Mit Monitoring
    - Welche Last wird abgearbeitet, wie sind die Antwortzeiten?
    - Wie ausgelastet sind die konfigurierten Softwarekomponenten?
      - Pools, Caches
    - Wie verhält sich die Garbage Collection?
  - Mit Java Thread Dumps
    - Warten wir auf andere Systeme (Backend, Datenbank, ...)?
    - Warten wir auf Locks (Software-Design)?
    - Loopen wir im Code?

© 2010 kippdata informationstechnologie GmbH

- Aufwärmrunde
- Motivation
- Java Management Extensions (JMX)
- JDK-Tools Java 5 und 6
- Thread Dumps
- Diskussion!

Java Management Extensions

- Java Management Extensions (JMX)
  - Sehr gute Möglichkeit interne Applikationszustände von außen abfragbar zu machen
    - Größen (Pools etc.), Zähler (Anfragen, Dauern, Fehlerzahlen)
    - komplexe Datenstrukturen
    - Aber auch Konfigurationseinstellungen
  - Operationen möglich
    - Reset, Resizing, Loglevel ändern, Aktivierung/Deaktivierung, ...
  - Notifications (Emitter, Listener)
    - Ereignismeldung, z.B Schwellwertüberwachung

© 2010 kippdata informationstechnologie GmbH

Java Management Extensions

- Java Management Extensions (JMX)
  - Java-Standard
    - JMX Version 1.4 (Java 6, ohne optionale Connector-Teile)
      - JSR-3, Standardbestandteil ab Java 5, vorher etwa MX4J
      - JSR-160, Java Management Extensions Remote API
    - JSR-262 Web Services Connector for JMX Agents: Java 7?
      - Bislang aber nur „echte“ Web Services zu sehen
    - JSR-255 (JMX 2.0): verschoben auf Java 8!

© 2010 kippdata informationstechnologie GmbH

Java Management Extensions

- Java Management Extensions (JMX)
  - Bereitstellung der Informationen in Form von MBeans
    - In jeder Java Laufzeitinstanz liegen einige MBeans vor
    - Viele Container bringen weitere MBeans mit
    - Es ist nicht schwierig MBeans selbst bereitzustellen
  - Zentrale Registrierung am MBeanServer
    - die MBeans bekommen eindeutige Namen (ObjectName)
    - meist nur eine MBeanServer pro JVM
  - Remote Management (Connector, Adapter)

© 2010 kippdata informationstechnologie GmbH

Java Management Extensions

- Zugriff von außen auf den MBeanServer
  - Seit Java 6: „Attach on Demand“
    - Lokaler Zugriff ohne Vorbereitung (Rechte auf Prozess nötig)
  - Für Zugriff über Netz werden System Properties gesetzt
    - http://java.sun.com/javase/6/docs/technotes/guides/management/agent.html#gdevf
    - -Dcom.sun.management.jmxremote \
      -Dcom.sun.management.jmxremote.port=9876 ...
    - Wichtig: in Produktion bitte mit Zugangsschutz!
    - Firewalls nicht unproblematisch (RMI macht weitere Ports auf)
      - Lösung JMXServiceURL und JMXConnectorServer
      - Beispiel: JmxRemoteLifecycleListener.java seit Tomcat 6

© 2010 kippdata informationstechnologie GmbH

Java Management Extensions

- JMX-Clients
  - JDK-Tools als Beispiele
  - Zugang über andere Konnektoren
    - Beispiele für Zugang via HTTP
      - JMXProxy in Tomcat
      - HTTP-Konnektor in MX4J
      - eigenes Servlet
  - jmxterm, Jmx4Perl, …
  - Nagios, Munin, ...: all have JMX plugins

Java Management Extensions

- Kleine Demo …
  - Zugang via JConsole
    - Targets:
      - Sleep.java
      - JConsole selbst
      - Tomcat
  - Zugang via Browser
    - Targets:
      - Tomcat Manager/JMXProxy
      - Webapp kpdtexplorer

© 2010 kippdata informationstechnologie GmbH

Java Management Extensions

- Beispiele Tomcat
  - Thread Pools: Auslastung
  - Datenbank-Pools: Auslastung
  - Gesamtlast Requests, Verarbeitungsdauer Requests
    - Ableitung: Last/Durchsatz und mittlere Antwortzeit im letzten Messintervall
  - Anzahl Sessions pro Webapp
- Aber auch
  - Aktuell in Bearbeitung befindliche URLs und bisherige Verarbeitungsdauer dieser Requests

© 2010 kippdata informationstechnologie GmbH

Java Management Extensions

- Auswertung
  - Es reicht nicht aus, interaktive Beobachtungen durchzuführen
    - Keine Historie, jeder User pollt für sich
    - ist aber häufig ein guter Einstieg!
  - Also: geeignete
    - geeignete Werkzeuge zum pollen und protokollieren auswählen
    - Schwellwerte festlegen
    - Auswerteverfahren aufsetzen (Visualisierung)

- Problembereiche
  - Skalare Attribute versus MXBeans (OpenMBeans)
    - Es werden zunehmend MBeans verwendet, die geschachtelte Daten zurückliefern
    - MXBeans erlauben es die Struktur der Daten herauszufinden
    - Viele Tools können das aber noch nicht
      - Wird zunehmend zu einem Auswahlkriterium
    - Deshalb soweit sinnvoll bei eigenen MBeans skalare Attribute verwenden

© 2010 kippdata informationstechnologie GmbH

- Problembereiche
  - MBeans orientieren sich häufig an der Source Code-Struktur
    - Nicht immer optimale Granularität
      - Viele MBeans des gleichen Typs
      - Dies erhöht enorm die Poll-Last
      - Beispiel Webcontainer: Verwende kleine Webapp, die die gewünschten Datensätze in der richtigen Granularität liefert
    - Sie liefern nicht immer exakt die Daten, die wir sehen wollen, sondern die, die intern gerade vorliegen
      - Beispiel: Maximale Größe Pool und aktuelle Größe Pool statt prozentuale Auslastung
      - Einfache Ableitungen von Größen nötig (Quotienten, Differenzen, Quotienten von Differenzen)

© 2010 kippdata informationstechnologie GmbH

Java Management Extensions

- Problembereiche
  - Das richtige Maß halten
    - Es gibt häufig zu viele MBeans
    - Was bedeuten die einzelnen MBean-Attribute?
    - Was wollen wir warum sehen?
    - Was fangen wir mit den Werten an?
      - Gibt es Schwellwerte für Gut/Schlecht?
      - Brauchen wir Aufzeichnungen für Auslastungsbetrachtungen (Sizing)

© 2010 kippdata informationstechnologie GmbH

- Aufwärmrunde
- Monitoring und Troubleshooting
- Java Management Extensions (JMX)
- JDK-Tools Java 5 und 6
- Thread Dumps
- Diskussion!

- JDK-Tools Java 5 und 6
  - Immer JDK verwenden
    - alle relevanten Tools sind nicht im JRE!
  - Die Tools werden mit Patchreleases verbessert
    - ab und zu mal ein JVM-Patchupdate wäre schön
  - Tools basieren auf Java, bekommen aber JVM-Optionen mittels „-J“
    - -J-d64
    - -J-mx32m

© 2010 kippdata informationstechnologie GmbH

JDK-Tools Java 5 und 6

- Tools Java5
  - jps: welche Javaprozesse laufen
  - jstat: laufende Textausgabe von Werten aus dem Plattform-MBeans
  - jconsole: GUI zur Anzeige von MBeans
- Kurze Demo
  - JConsole erweiterbar mit Plugins (Custom Tabs)
    - http://java.sun.com/javase/6/docs/technotes/guides/management/jconsole.html#gdeje

JDK-Tools Java 5 und 6

- Tools Java 6 (zusätzlich)
  - jinfo: Ausgabe und Manipulation von Flags
  - jstack: Ausgabe von Stack-Dumps
  - jmap & jhat: Dump and Analyse des Heaps
  - jvisualvm: mächtiges GUI, erweiterbar durch Plugins
- Kurze Demo
  - JVisualVM erweiterbar mit Plugins
    - https://visualvm.dev.java.net/
    - Dort auch aktuellere Version verfügbar!

© 2010 kippdata informationstechnologie GmbH

JDK-Tools Java 5 und 6

- Diese Tools sind sehr gut, um sich einzuarbeiten
  - Welche Daten gibt es?
  - Wie verhalten sie sich?
  - Wie hängen sie zusammen?
  - Was davon interessiert mich?

  und die Kreativität anzuregen!
- Oder auch um nicht vorbereitete Fragen ad hoc anzugehen
  - Das ist manchmal nötig

JDK-Tools Java 5 und 6

- Diese Tools sind nicht geeignet für den dauerhaften Einsatz im Enterprise-Umfeld
  - Typische Anforderungen
    - dauerhaftes Sammeln von Daten
    - automatisierte Auswertung
      - Schwellwerte, Verdichtung, Visualisierung, Trends
      - Ergebnisse sollen persistent sein
    - zentrales Daten-Repository
    - zentrale Konfiguration
      - Anwendungs- und Container-Typen, Farmknoten, MBeans

© 2010 kippdata informationstechnologie GmbH

- Aufwärmrunde
- Monitoring und Troubleshooting
- Java Management Extensions (JMX)
- JDK-Tools Java 5 und 6
- Thread Dumps
- Diskussion!

© 2010 kippdata informationstechnologie GmbH

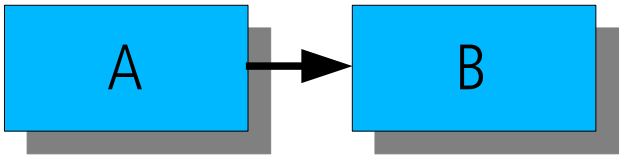
Thread Dumps

- Problembeschreibung
  - Performanceprobleme
    - Anwendung ist zu langsam (einzelne Vorfälle)
    - Es bilden sich Staus aus
  - und dann meist als Folge Stabilitätsprobleme
    - Anwendung reagiert nicht mehr
  - Fragen
    - Wo kommen die Performance-Probleme her?
      - Warten auf Locks, Fremdsysteme?
      - Intensive Berechnungen?
    - Wo kommt ein zu hoher CPU-Verbrauch her (bei Extremen)?

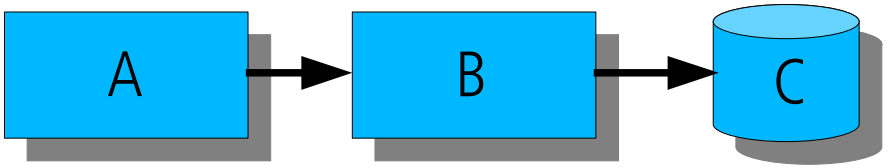
© 2010 kippdata informationstechnologie GmbH

Thread Dumps

- Performance-Probleme bei verteilten Systemen
  - Die Ursache ist nicht meine Komponente A, es ist wohl B



  - Die Ursache ist nicht meine Komponente B, es ist wohl C



  - Die Datenbank C langweilt sich
  - Dann ist es wohl das Netz!
  - Die Erfahrung zeigt: meistens nicht.

Thread Dumps

- Inhalt eines Thread-Dumps
  - Ein Thread-Dump ist eine Momentaufnahme der Code-Ausführung in der JVM
  - Er enthält eine Liste aller Threads in der JVM
    - Mit Name und ID, sowie Zustand (etwa runnable)
      - ID meist abbildbar auf die Thread-Nummern des OS
    - Mit komplettem Funktionsstack der Java-Methoden
    - Mit Informationen bzgl. des Wartens auf Locks
    - Mit Ausgabe, ob ein Deadlock vorliegt

Thread Dumps

- Momentaufnahme
  - Meist muss mehr als ein Thread-Dump gemacht werden, um Zufallsbeobachtungen auszuschliessen
    - Z.B. 3 Dumps im Abstand von jeweils 3 Sekunden
- Thread-Dumps sind ein JVM-Feature
  - Klappt also für alle Java-Prozesse!
- Thread-Dumps gehen sehr schnell
- Thread-Dumps sind OK in Produktion! (ca. ab 1.4.2_10)
- Auch regelmäßig (alle 10 Minuten; nicht: Sekunden)

© 2010 kippdata informationstechnologie GmbH

Thread Dumps

- Tipp: Wegschreiben von 3 Thread Dumps ins Stopp-Skript aufnehmen
  - Gerade unter Stress wird nicht daran gedacht vor dem Neustart noch Dumps zu machen
- Wohin geht der Dump?
  - Nach STDOUT
  - Im Startskript auffangen
    - Zeitstempel hinzufügen
    - Rotieren

Thread Dumps

- Wie wird ein Dump erzeugt?
  - Unix/Linux: sende QUIT-Signal an Prozess
    - Das darf der root-User und der Owner des Prozesses
    - Es beendet **nicht** den Prozess! Der Name ist irreführend.
  - Windows: send Break-Signal an den Prozess
    - Das darf nur ein Prozess, der in der gleichen Console-Gruppe wie die JVM ist
    - Meist nur auf Entwickler-PC oder in speziellen Fällen in Produktion so machbar
      - Start über DOS-Box
      - Demo: Tomcat

Thread Dumps

- Was machen wir bei einem Windows-Service?
  - Hat kein Terminal, deshalb klappt Break-Signal nicht
- Ab Java 5 gibt es einen programmatischen Weg, aus der Java-Anwendung heraus Thread-Dumps aufzurufen
  - **java.lang.management.ThreadMXBean**
  - Der Dump ist etwas weniger aussagekräftig
    - Weniger Lock-Information (besser ab Java 6)
    - Weniger IDs zum Thread
    - Keine JVM-internen Threads

© 2010 kippdata informationstechnologie GmbH

Thread Dumps

- Aufrufmöglichkeiten MBean-basierter Thread-Dump
  - Aufrufbar über JMX-Schnittstelle
    - Ab Java 6 ohne vorherige Aktivierung der Schnittstelle (Attach on Demand)
    - jstack (kann aber keine Authentisierung)
    - <JDK_HOME>/demo/management/FullThreadDump/FullThreadDump.jar
    - Eigener Client
  - Oder kapseln in HTTP-Servlet oder ähnlichem Aufrufweg
    - Demo: kpdtexplorer

© 2010 kippdata informationstechnologie GmbH

Thread Dumps

- # Ausgabe Deadlock

```
Deadlock found :-
"ajp-127.0.0.1-8009-42" Id=486 in
     BLOCKED on lock=java.lang.Object@1d461d7
      owned by ajp-127.0.0.1-8009-38 Id=460
    at jcifs.smb.SmbTree.treeConnect(SmbTree.java:128)
    at jcifs.smb.SmbTree.send(SmbTree.java:64)
    at jcifs.smb.SmbTree.treeDisconnect(SmbTree.java:168)
    at jcifs.smb.SmbSession.logoff(SmbSession.java:301)
    at jcifs.smb.SmbTransport.getSmbSession(SmbTransport.java:138)
    at jcifs.smb.SmbSession.logon(SmbSession.java:167)
    at jcifs.smb.SmbSession.logon(SmbSession.java:162)
...
"ajp-127.0.0.1-8009-38" Id=460 in
     BLOCKED on lock=jcifs.smb.SmbTransport@14cfb2c
      owned by ajp-127.0.0.1-8009-42 Id=486
    at jcifs.smb.SmbTree.treeConnect(SmbTree.java:130)
    at jcifs.smb.SmbSession.logon(SmbSession.java:169)
    at jcifs.smb.SmbSession.logon(SmbSession.java:162)
...
```

© 2010 kippdata informationstechnologie GmbH

Thread Dumps

- Konsequenz Deadlock
  - Über 400 Threads im Stack

```
"ajp-127.0.0.1-8009-750" Id=2382 in
     BLOCKED on lock=jcifs.smb.SmbTransport@14cfb2c
      owned by ajp-127.0.0.1-8009-42 Id=486
    at jcifs.util.transport.Transport.connect(Transport.java:151)
    at jcifs.smb.SmbTransport.connect(SmbTransport.java:287)
    at jcifs.smb.SmbSession.getChallenge(SmbSession.java:146)
    at jcifs.smb.SmbSession.getChallenge(SmbSession.java:140)
...
```

  - Entstanden innerhalb weniger Minuten
  - Keine Neuanmeldung mehr möglich
  - Obwohl nur zwei Threads im Deadlock sind, bleiben sehr viele Threads dahinter hängen

© 2010 kippdata informationstechnologie GmbH

Thread Dumps

- Entwickler-Kommentar zu diesem Deadlock

„I believe the deadlock could only be triggered by a contrived test that creates an excessive number of threads that would never been used in a normal application."

  - dict.leo.org:
    contrived = arrangiert, erfunden, gekünstelt, gestellt
  - Hier war es normale Produktion!

Thread Dumps

- Lock-Contention
  - Viele Threads im Stack

```
 "ajp-127.0.0.1-8009-XXX" Id=YYY in TIMED_WAITING on
lock=com.mybiz.myapp.webapp.handler.FormHandler@be635d
     at java.lang.Object.wait(Native Method)
     at com.mybiz.myapp.webapp.handler.FormHandler.lock(Unknown Source)
     at com.mybiz.myapp.webapp.UpdateAction.execute(Unknown Source)
     at com.mybiz.myapp.webapp.ActionBase.execute(Unknown Source)
     at org.apache.struts.action.RequestProcessor.
        processActionPerform(RequestProcessor.java:484)
     at org.apache.struts.action.RequestProcessor.
        process(RequestProcessor.java:274)
     at org.apache.struts.action.ActionServlet.process(ActionServlet.java:1482)
     at org.apache.struts.action.ActionServlet.doPost(ActionServlet.java:525)
 ...
```

  - Die Thread-IDs XXX und YYY wechseln langsam im Laufe der Zeit

Thread Dumps

- Dump-Auswertung
  - Meist sehen wir uns zunächst nur die obersten 5-10 Methoden des Stacks an
  - Wir wollen alle Threads gruppieren, deren Top-N Methoden-Stack gleich ist
    - Was machen die meisten Threads gerade?
  - Hier bietet sich ein Auswerteskript an
  - Bei Ideen, was merkwürdig sein könnte, immer die Idee im vollen Dump überprüfen
  - Demo Stack-Statistik im kpdtexplorer

Thread Dumps

- Phänomene, die sich meist gut durch Thread-Dumps verstehen lassen
  - Die Anwendung ist insgesamt langsam, obwohl die CPU-Auslastung gering ist
    - Dann wird meistens auf etwas gewartet
      - Remote Calls (Middleware, DB, Webservices)
      - Locks
    - Lieblingsformel
      Durchsatz * Antwortzeit = Parallelität
    - Normale Anfragerate, erhöhte Antwortzeit => erhöhte Threadzahl

© 2010 kippdata informationstechnologie GmbH

Thread Dumps

- Phänomene, die sich meist gut durch Thread-Dumps verstehen lassen – Fortsetzung
  - Die CPU-Auslastung ist zu hoch
    - Was ist auf der CPU (Thread-Nummer)
    - Was macht der Thread?
  - Die Anwendung reagiert nicht mehr
    - Deadlock?
    - Alle Threads warten auf remote?

Thread Dumps

- Vorschlag
  - Trauen Sie Sich Thread-Dumps in der Produktion zu machen
    - Natürlich erst im Test bzw. Staging
    - Bei wichtigen Anwendungen auch proaktiv und regelmässig
  - Betrieb und Entwicklung sollten zusammen versuchen die Dumps zu verstehen
    - Insbesondere auch im Gut-Fall
  - Lassen Sie Sich nicht von der Größe abschrecken
    - Skript zum Zusammenfassen gleicher Top-N-Stacks

© 2010 kippdata informationstechnologie GmbH

- Aufwärmrunde
- Motivation
- Java Management Extensions (JMX)
- JDK-Tools Java 5 und 6
- Thread Dumps
- Diskussion!

© 2010 kippdata informationstechnologie GmbH

Diskussion

- Fragen?

- Abspann

kippdata – Kontakt

- Wie können wir Ihnen helfen?
- So erreichen Sie uns:
  - info@kippdata.de
  - rainer.jung@kippdata.de
  - www.kippdata.de
  - 0228/98549-0

Abspann

- kippdata informationstechnologie GmbH
  - Professioneller Open Source Support
    - Schwerpunkt Apache Tomcat und Apache httpd
  - Betriebskonzepte zu Hochlast und Ausfallsicherung
  - Healthchecks von produktiven JEE-Anwendungen
  - Lasttests und Sizing, Analyse von Produktionsproblemen
  - Workshops
- Besuchen Sie uns auf www.kippdata.de, und www.kippdata.de/tomcat